AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

Please cancel claim 1 of the original parent application.

2. (new) A compiler, disposed on a computer readable medium, the compiler including instructions to cause a processor to:

access source code expressed in a computer language having an instruction set that includes:

an instruction set statement to declare a network protocol, the statement having a syntax to name packet data at one or more specified locations within a network packet; and

an instruction set statement to specify a rule, the rule syntax comprising a Boolean expression and at least one action to perform if the Boolean expression is true; and

based on the received source code, output instructions to cause a processor to process received packets by:

assigning value(s) to the named packet data based on data within a packet in accordance with the instruction set statement to declare a network protocol; and

applying at least one rule statement.

3. (new) The compiler of claim 2, wherein the source code comprises multiple rules and where the output instructions cause the processor to apply the rules in an

order corresponding to the order in which the rule statements appear in the source code.

4. (new) The compiler of claim 3, wherein the at least one action of a rule statement returns a value controlling whether subsequent rules are applied to the packet.

5. (new) The compiler of claim 2,

wherein the instruction set statement to declare a network protocol comprises a statement having a syntax comprising of:

protocol protocol_name { [field definitions(s)] }

and wherein the syntax to name packet data comprises field definitions having a syntax comprising of:

field_name {protocol_name [offset : field size] }.

6. (new) The compiler of claim 2,

wherein the syntax of the instruction set statement to declare a network protocol further comprises at least one declaration of an encapsulated packet header associated with another network protocol; and

wherein the output instructions comprise instructions to name data of the encapsulated packet header.

7. (new) The compiler of claim 6, wherein the at least one declaration of the encapsulated packet header comprises a declaration including an evaluation expression identifying whether the encapsulated packet header is present in a packet.

8. (new) The compiler of claim 7, wherein the declaration of the encapsulated packet header comprises a declaration having a syntax comprising of:

demux { Boolean_expression {protocol_name at offset} }

wherein the protocol_name identifies a protocol declared by an instruction statement to declare a network protocol.

9. (new) The compiler of claim 2, wherein the instruction set includes:

an instruction set statement to name a set of values, the set of values comprising a set of tuples, wherein each tuple comprises one or more key values; and

an instruction set search statement to search the set of values.

10. (new) The compiler of claim 9, wherein the instruction set search statement comprises a syntax of:

set_name.search_name(key(s)),

wherein the search_name is an expression that evaluates to true when at least one tuple of the set of values identified by the set_name matches the value(s) of the key(s) specified.

11. (new) The compiler of claim 9, wherein the set of values are set by instructions external to the instructions output in response to the source code.

12. (new) The compiler of claim 11, wherein the values are set based on received packets.

13. (new) A method, comprising:

accessing source code expressed in a computer language having an instruction set that includes:

an instruction set statement to declare a network protocol, the statement having a syntax to name packet data at one or more specified locations within a network packet; and

an instruction set statement to specify a rule, the rule syntax comprising a Boolean expression and at least one action to perform if the Boolean expression is true; and

based on the received source code, outputting instructions to cause a processor to process received packets by:

assigning value(s) to the named packet data based on data within a packet in accordance with the instruction set statement to declare a network protocol; and

applying at least one rule statement.

14. (new) The method of claim 13, wherein the source code comprises multiple rules and where the output instructions cause the processor to apply the rules in an order corresponding to the order in which the rule statements appear in the source code.

15. (new) The method of claim14, wherein the at least one action of a rule statement returns a value controlling whether subsequent rules are applied to the packet.

16. (new) The method of claim 13,

wherein the instruction set statement to declare a network protocol comprises a statement having a syntax comprising of:

protocol protocol_name { [field definitions(s)] }

and wherein the syntax to name packet data comprises field definitions having a syntax comprising of:

field_name {protocol_name [offset : field size] }.

17. (new) The method of claim 13,

wherein the syntax of the instruction set statement to declare a network protocol further comprises at least one declaration of an encapsulated packet header associated with another network protocol; and

wherein the output instructions comprise instructions to name data of the encapsulated packet header.

18. (new) The method of claim 17, wherein the at least one declaration of the encapsulated packet header comprises a declaration including an evaluation expression identifying whether the encapsulated packet header is present in a packet.

19. (new) The method of claim 18, wherein the declaration of the encapsulated packet header comprises a declaration having a syntax comprising of :

demux { Boolean_expression {protocol_name at offset} }

wherein the protocol_name identifies a protocol declared by an instruction statement to declare a network protocol.

20. (new) The method of claim 13, wherein the instruction set includes:

an instruction set statement to name a set of values, the set of values comprising a set of tuples, wherein each tuple comprises one or more key values; and

an instruction set search statement to search the set of values.

21. (new) The method of claim 20, wherein the instruction set search statement comprises a syntax of:

set_name.search_name(key(s)),

wherein the search_name is an expression that evaluates to true when at least one tuple of the set of values identified by the set_name matches the value(s) of the key(s) specified.

22. (new) The method of claim 21, wherein the set of values are set by instructions external to the instructions output in response to the source code.

23. (new) The method of claim 22, wherein the values are set based on received packets.